Рекомендації з написання PHP коду

Вітаю! Сьогодні розглянемо основні рекомендації щодо написання чистого та ефективного коду PHP. Ці рекомендації допоможуть вам писати код, який легко читати, зрозуміти та підтримувати.

Шеремет Андрій Григорович
ПЗПІ-22-6

13.12.2024

Мета презентації

Дослідження основних рекомендацій написання коду для мови програмування PHP з метою покращення якості, читабельності та підтримуваності програмного забезпечення, а також забезпечення відповідності стандартам PSR (PHP Standards Recommendations).

Що таке Code Conventions?

Code Conventions - це набір правил та рекомендацій щодо написання коду. Вони охоплюють стиль написання, іменування змінних, функцій та класів, а також форматування коду.

1 **Стиль коду**

Відступи, пробіли, розбиття на рядки.

2 **Іменування**

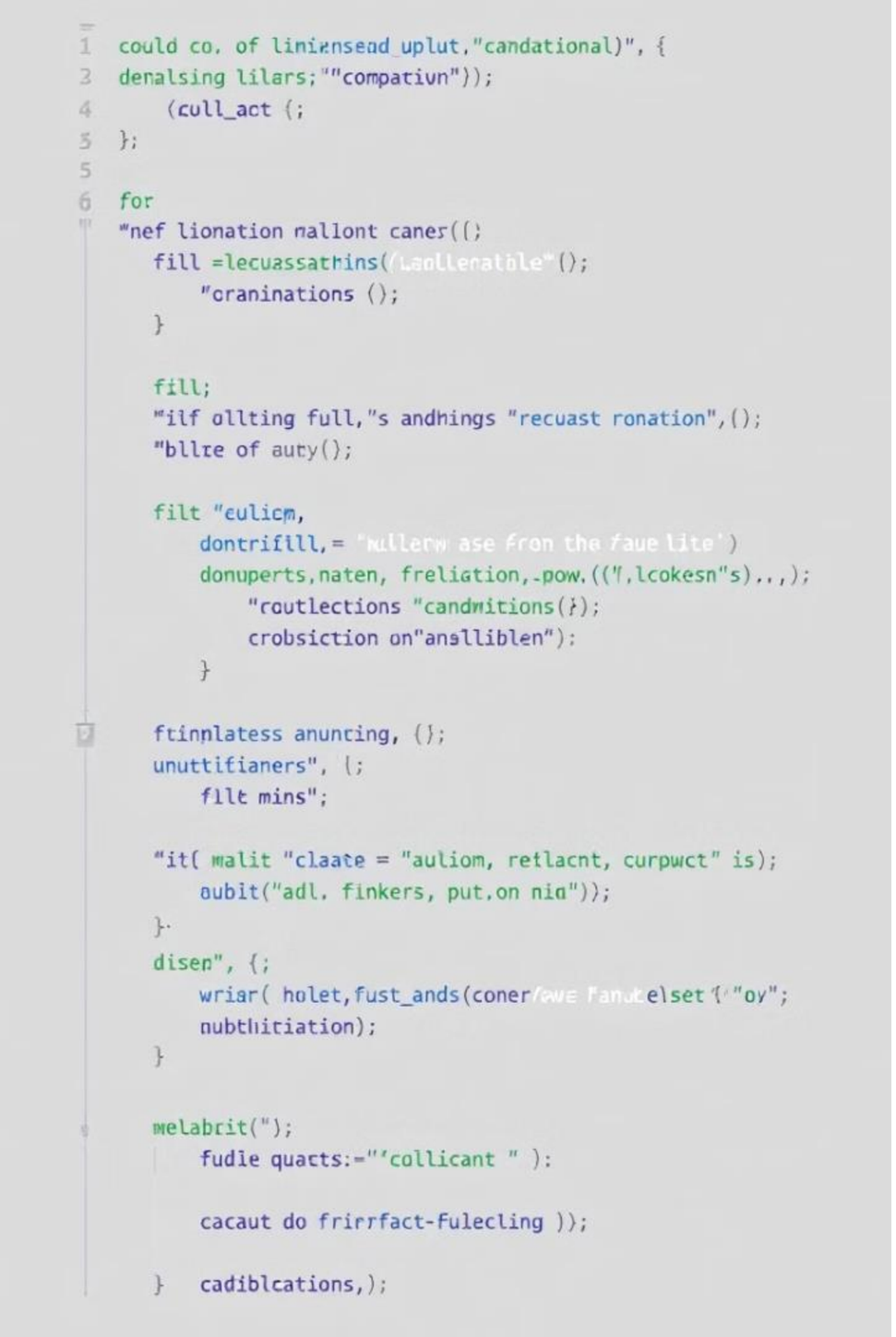
Вибір назв для змінних, функцій та класів.

3 **Коментарі**

Які коментарі додавати до коду та як їх формулювати.

4 **Організація**

Структурування коду на файли та папки.



Чому важливо дотримуватися Code Code Conventions?

Дотримання Code Conventions покращує якість та зрозумілість коду, що спрощує його читання, підтримку та модифікацію.

Зменшення помилок

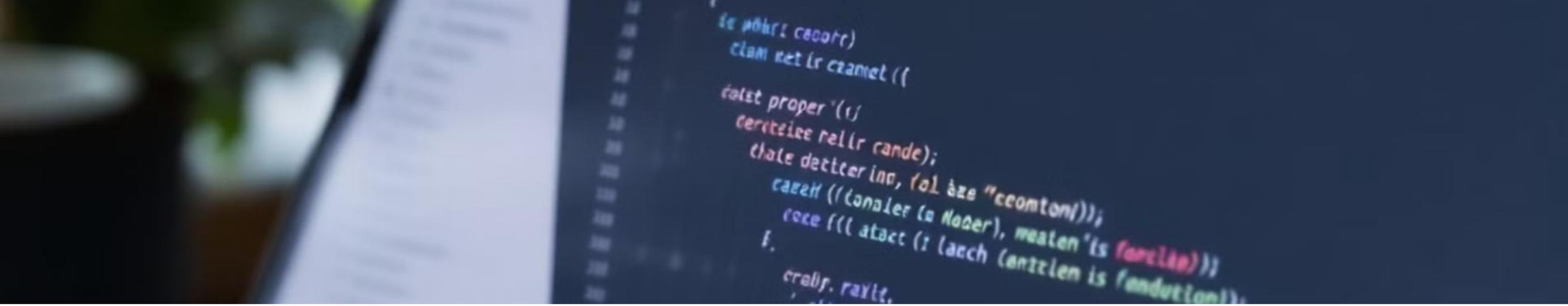
Стандартний стиль коду спрощує процес написання коду та зменшує кількість помилок.

Збільшення читабельності

Чітка структурованість коду спрощує його розуміння для інших розробників.

Зручне обслуговування

Послідовний код легше підтримувати та змінювати в майбутньому.



Основні рекомендації

Розглянемо основні рекомендації, які допоможуть вам написати якісний та ефективний код PHP.

Стиль коду

1 Відступи

Використовуйте 4 пробіли для відступів, не табуляцію

2 Дужки та пробіли

Після ключових слів **if, for, while, switch, function** обов’язково ставте пробіл перед дужками

Відкриваюча дужка { для функції, класів, методів повинна бути на тій самій лінії, що й декларація

3 Довжина рядка

Максимальна довжина рядка — 120 символів. Для кращої читабельності рекомендовано обмежувати рядки до 80 символів.

Приклад коду:

```php
<?php

function example() {
    if ($condition) {
        echo 'Hello!';
    }
}
```

Іменування

1 camelCase для функцій

Використовуйте camelCase для імен змінних і функцій

2 PascalCase для класів

Використовуйте PascalCase для назв класів

3 Константи

Константи пишіть великими літерами з підкресленнями

Приклад коду:

```php
<?php

define('API_KEY', '12345');

class UserController {
    const MAX_LIMIT = 100;
    function getUserData() {
        $userName = 'John';
    }
}
```

Коментарі

1 Однорядкові коментарі

Використовуйте однорядкові коментарі для пояснення простого коду

2 PHPDoc

Для складних функцій додавайте блоки PHPDoc

3 Зрозумілість коментарів

Пишіть коментарі у зрозумілій і лаконічній формі, уникаючи очевидних пояснень.

Приклад коду:

```php
<?php

// Ініціалізація змінної
$count = 0;

/**
 * Підрахунок суми двох чисел.
 *
 * @param int $a
 * @param int $b
 * @return int
 */
function add($a, $b) {
    return $a + $b;
}
```

Організація коду

1 **Модулі**

Розбивайте код на модулі та класи для полегшення підтримки та повторного використання.

2 **1 - 1**

Дотримуйтеся принципу одна функція/метод — одна одна задача.

3 **Групування**

Групуйте логічно пов’язані функції та методи разом у класах або неймспейсах.

Приклад коду:

```php
<?php
namespace App\Controllers;

class UserController {
    public function create() {
        // ...
    }
    public function update() {
        // ...
    }
}
```

Приклад неправильно написаного коду

Неохайний код без відступів, неправильне іменування.

```php
<?php
const $usertoken;
class user_service {
function GetuserData($id){
$userdata=['id'=>$id,'name'=>'John Doe','email'=>'john.doe@example.com'];return $userdata;
}

function UPDATEuserdata($id,$data){
    // Логіка оновлення
return true;}
}
```

Приклад правильно написаного коду

Чистий код з відступами, чіткими назвами змінних.

```php
<?php
declare(strict_types=1);

namespace App\Services;

/**
 * Клас UserService для роботи з користувачами.
 */
class UserService
{
    /**
     * Отримати дані користувача за його ідентифікатором.
     *
     * @param int $userId Ідентифікатор користувача.
     * @return array Дані користувача.
     */
    public function getUserData(int $userId): array
    {
        // Імітація даних користувача
        $userData = [
            'id' => $userId,
            'name' => 'John Doe',
            'email' => 'john.doe@example.com',
        ];

        return $userData;
    }

    /**
     * Оновити дані користувача.
     *
     * @param int $userId Ідентифікатор користувача.
     * @param array $data Дані для оновлення.
     * @return bool Результат оновлення.
     */
    public function updateUserData(int $userId, array $data): bool
    {
        // Тут могла б бути логіка оновлення даних у базі
        // Повертаємо успішний результат
        return true;
    }
}
```

Роль Code Conventions у командній роботі

В команді важливо дотримуватися однакових правил написання коду. Це спрощує комунікацію, зменшує кількість помилок та покращує ефективність розробки.

1

Спільне розуміння

Всі члени команди розуміють структуру коду.

2

Легше обслуговувати

Простіше підтримувати та змінювати код у майбутньому.

3

Зменшення помилок

Спільне дотримання правил зменшує кількість помилок.

Висновки

Дотримання Code Conventions - важливий крок для створення якісного, зрозумілого та ефективного коду. Це покращує процес розробки та робить ваш код легшим для розуміння, підтримки та модифікації. Він включає в себе дотримання стилю, іменуванню коду та коментарів.

1

Чіткий код

Код, який легко читати та зрозуміти.

2

Ефективний код

Код, який працює швидко та без помилок.

3

Зручний код

Код, який легко підтримувати та змінювати.